基于多蚁群同步优化的多真值发现算法 *

冯　钦 [1]，曹建军 [2]，郑奇斌 [1]，张　磊 [1]，翁年凤 [2]，李红梅 [1]

(1. 陆军工程大学 指挥控制工程学院，南京 210007; 2. 国防科技大学 第六十三研究所，南京 210007)

摘　要：为提高在多真值场景下真值发现的准确性，提出一种多蚁群同步优化的多真值发现算法（multi-ant colonies synchronization optimization based multi-truth discovery algorithm，MAC-SO-MTD）。以最大化各数据源提供的观测值集合与该对象真值集合之间相似度的加权和为目标，将多真值发现问题建模为求解子集问题，在此基础上设计蚁群算法进行求解：根据对象个数设置相应的蚁群，构造子集问题的有向图，利用路径概率转移公式进行同步搜索真值；将信息素更新分为本次迭代最优更新和本次迭代不更新，提高了算法的收敛速度。最后，通过算法复杂度分析和在真实数据集上的实验验证了该算法的优越性。

关键词：数据清洗；数据冲突；多真值发现；子集问题；蚁群优化

中图分类号：TP311　　doi: 10.19734/j.issn.1001-3695.2018.05.0453

Multi-ant colonies synchronization optimization based multi-truth discovery algorithm

Feng Qin[1], Cao Jianjun[2], Zheng Qibin[1], Zhang Lei[1], Weng Nianfeng[2], Li Hongmei[1]

(1. *Command & Control Engineering College*, *Army Engineering University*, *Nanjing* 210007, *China*; 2. *The* 63*rd Research Institute*, *National University of Defense Technology*, *Nanjing* 210007, *China*)

Abstract: In order to improve the accuracy of truth discovery in multi-truth scene, this paper proposed a multi-ant colonies synchronization optimization based multi-truth discovery (MAC-SO-MTD) algorithm. It modeled the multi-truth discovery problem as the subset problem, which goal was maximizing the weighted sum of similarity between the set of observations provided by each data source and the set of true values of the object. On this basis, then designed ant colony algorithm to solve the problem. It set ant colonies according to the number of objects. Based on the subset problem's structure graph, this paper used routes' probability transition equations to search for truths synchronically. After one cycle, the best route of this cycle updating and no updating were two instances of updating pheromone, which improved the convergence speed. Finally, the analysis of algorithm complexity and contrast experiment on the real data set validated the superiority of the algorithm.

Key words: data cleaning; data conflict; multi-truth discovery; subset problem; ant colony optimization

0　引言

随着大数据时代的到来，互联网的高速发展和产业的数字化导致各种数据量急剧增长，同时也带来各种数据质量问题。由于互联网的开放性和多源特性，不同互联网平台提供的数据参差不齐，所以网络上的数据不一定都是真实的，错误、过时、不完整等数据的存在会导致多个数据源对同一实体的描述存在着冲突[1]。例如，不同天气网站针对某一地方提供不同的天气情况；不同购物网站为同一商品提供了不一致的产品信息等。根据“垃圾进，垃圾出（garbage in, garbage out）”的原理可知，低质量的冲突数据可能导致错误的分析决策和预测，对于相关信息产业的影响十分巨大[2]。因此，解决数据冲突问题格外关键且迫在眉睫。

Yin 等人[3]针对冲突处理问题首先定义了真值发现问题，即给定多个数据源提供的对于多个真实对象的大量冲突描述信息，研究如何从这些冲突信息中为每一个真实对象找出最准确的描述。有些对象在不同数据源中只对应一种描述，即只有一个真值，如一个人只有一个身份证号，这类单值属性对应的真值发现问题为单真值发现问题；有些对象在不同数据源中对应多种描述，即存在多个真值，如一本书可以有多个作者、一个人可以有多个头衔等，这类多值属性对应的真值发现问题为多真值发现问题。

收稿日期：2018-05-21；**修回日期：**2018-07-13　　**基金项目：**国家自然科学基金资助项目（61371196）

作者简介：冯钦（1993-），男，广东徐闻人，硕士研究生，主要研究方向为数据质量控制与冲突消解；曹建军（1975-），男（通信作者），副研究员，博士，主要研究方向为数据质量控制与治理、数据智能分析(jianjuncao@yeah.net)；郑奇斌(1990-)，男，博士研究生，主要研究方向为数据工程；张磊(1989-)，男，硕士研究生，主要研究方向为数据工程；翁年凤（1983-），男，工程师，博士，主要研究方向为数据工程、软件工程；李红梅（1990-），女，博士研究生，主要研究方向为个性化推荐、数据挖掘.

多真值发现问题的求解与单真值发现问题不同。文献[3~9]针对单真值问题，提出"对象真值唯一"的假设，并选取对象属性中得分最高或概率最大的值作为真值。而多真值发现问题不但要找到正确的值，还要尽可能地将所有的真值都找到。文献[10~12]提出的方法可以解决多真值发现问题。但文献[10]需要设置阈值来选择真值集合，文献［11］不适用于各数据源均仅提供了部分实体真值的情况，文献［12］需考虑数据源中对象的领域信息。

马如霞等人[13]提出了 MTruths 算法，能处理各数据源均仅提供了部分实体真值的情况，可以直接返回对象的真值集合，避免了阈值的选择问题，且其准确性优于已有多真值算法。但当对象真值集合基数较大时，MTruths 算法中基于贪心策略的方法容易陷入局部最优，降低多真值发现的准确性。本文针对这一问题，通过将多真值发现过程转换为求解子集问题，并设计蚁群算法同步进行多真值发现，在对象真值集合基数较大时能较好地进行多真值发现。

本文的主要贡献如下:

a) 将多真值发现过程转换为求解子集问题，通过最大化各数据源提供的观测值集合与对象真值集合之间相似度的加权和，在给定的对象值集中选出合适的真值集合，避免了通过设置阈值来选择真值;

b) 设计蚁群算法求解问题，根据对象数量设置相应的蚁群同步进行多真值发现，能在合理时间内找到较优解;

c) 通过真实数据集上的实验验证了本文提出的算法的优越性。

1　相关工作

针对单真值问题，研究者们进行了大量的研究。文献[3]首先提出了真值发现的概念，并根据链路分析的思想提出了 TruthFinder 算法。文献[4]基于概率投票的迭代计算方式提出了 IVote 算法，并在此基础上考虑数据源的权威性，即数据源的投票比重提出了 IRVote 算法。文献[5～7]针对具有不同数据类型的多源数据，考虑了异构数据的真值发现问题。文献[8]考虑数据源复制和数据的复制关系，通过一个多层概率模型提高了 Web 数据的可用性。文献[9]针对数据源间可能存在的数据复制问题，将特定于每条事实的联合召回率和联合假真率融入真值概率计算。

上述这些方法都是基于真值唯一的假设，通过选择对象属性中得分最高或概率最大的值作为真值。对于多真值的情况，大部分算法模型并不适用。

针对多真值发现问题，文献[10]首先提出了可以处理多值属性真值发现的方法（latent truth model, LTM），但该方法假设数据源的查全率和查准率服从 Beta 分布，如果真实数据集不满足假设的分布，则会对效果造成很大影响。文献[11]借鉴 HITS（hypertext-induced topic search）算法思想，提出了多真值迭代发现算法，将数据源为实体提供的描述集看做实体在数据源上的视图，定义视图链接关系图，依据视图与描述相互迭代影响计算，但该算法并不适用各数据源均仅提供了部分实体真值的情况。文献［12］考虑数据源中不同领域对象的多真值发现问题，提出了一种集成贝叶斯的方法来考虑数据源内各领域对象描述的可信度，能够不需要任何监督来进行多真值发现，但该算法要求对象需具备其对应的领域信息。文献［13］提出了 MTruths 算法，将多真值发现问题转换为一个最优化问题，并在真值计算过程中采用了基于枚举的方法和基于贪心策略的方法。该算法可以直接得到对象的真值集合，避免通过阈值的设置选择对象真值。但当对象真值集合基数较大时，MTruths 算法的准确性较低，而本文提出的方法能较好地处理对象真值集合基数较大时的多真值发现问题。

2　问题描述

多真值发现问题假设对象的真值是一个集合。表 1 列举了提供《Distributed Systems: Concepts and Design》一书的五个网站及其提供的作者信息。

表 1　Distributed systems: concepts and design 作者信息

Table 1　Author information of discreted systems: concepts and design

网站	作者
happybook	Coulouris George F; Dollimore Jean; Kindberg Tim
EnjoyStudy	Coulouris; Dollimore Jean; Kindberg Tim
Sunmark Store	Coulouris
The Book Depository	George Coulouris
Books2Anywhere.com	Coulouris George F; Dollimore Jean; K

由表 1 可知，每个网站提供的作者信息都不一样且难以判断真假，当想要收集这些信息时就存在一定困难。多真值发现就是要从这些多源冲突数据中发现真值集合。

给定对象集合 $O=\{o_1, o_2, \ldots, o_k, \ldots, o_n\}$，其中 n 是对象总数，o_k 表示第 k 个对象。数据源集合 $S=\{s_1, s_2, \ldots, s_h, \ldots, s_m\}$，数据源提供对象描述信息，$s_h$ 表示第 h 个数据源，其中 m 是数据源总数。对象 o_k 可以有多个真值，数据源 s_h 可以为其提供观测值的集合。对象 o_k 的观测值集 $V_{*,k}=\{v_1, v_2, \ldots, v_{Lk}\}$，表示所有数据源对对象 o_k 提供的观测值的集合，其中 L_k 是对象 o_k 观测值集的基数。对象 o_k 根据算法求解得到的真值集合可表示为 $V'_{*,k}$，$V'_{*,k}$ 是观测值集 $V_{*,k}$ 的子集。

本文研究的问题为：给定数据源集合 $S=\{s_1, s_2, \ldots, s_h, \ldots, s_m\}$，对其描述的对象集合 $O=\{o_1, o_2, \ldots, o_k, \ldots, o_n\}$，根据每个对象 o_k 的观测值集 $V_{*,k}$ 找出其所有真值集合 T_k。

3　多真值发现模型

本章首先介绍多真值发现算法的模型，然后对该模型进行分析。

3.1　模型概述

根据两个假设：a)对象的真值情况应该尽可能与各数据源

提供的观测值接近；b)数据源的质量越高则其提供的对象属性集合与真值集合越相似[13]。因此可将多真值发现问题建模如式(1)~(3)所示。

$$\max\ \Phi' = \sum_{k=1}^{n}\sum_{h=1}^{m} w_h \times f(V_{h,k}, V'_{*,k}) \tag{1}$$

$$s.t.\ \sum_{h=1}^{m} w_h = 1 且 w_h \in [0,1] \tag{2}$$

$$V'_{*,k} \subseteq V_{*,k} \tag{3}$$

模型以各对象的真值集合和数据源提供的该对象观测值集合之间相似度的加权和达到最大为目标（式(1)）。式(1)中：w_h 为数据源 s_h 的数据质量权重；$V_{h,k}$ 为数据源 s_h 为对象 o_k 提供的观测值集合；$V'_{*,k}$ 为算法得到的对象 o_k 的真值集合，且是 $V_{*,k}$ 的子集；$f(V_{h,k}, V'_{*,k})$ 定义为集合 $V_{i,k}$ 与 $V'_{*,k}$ 的 Jaccard 相似度。

$$f(V_{h,k}, V'_{*,k}) = \frac{|V_{h,k} \cap V'_{*,k}|}{|V_{h,k} \cup V'_{*,k}|} \tag{4}$$

w_h 定义如下：

$$w_h = \frac{\sum_{k=1}^{n} f(V_{h,k}, V'_{*,k})}{\sum_{h=1}^{m}\sum_{k=1}^{n} f(V_{h,k}, V'_{*,k})} \tag{5}$$

式(5)中：$\sum_{k=1}^{n} f(V_{h,k}, V'_{*,k})$ 为数据源 s_h 内所有对象的相似度之和；$\sum_{h=1}^{m}\sum_{k=1}^{n} f(V_{h,k}, V'_{*,k})$ 表示所有数据源内所有对象的相似度之和。

3.2　模型分析

由于式(1)中每个项目中的真值相互独立，所以当每个对象的真值集合与对应的观测值集合相似性达到最大时，式(1)即可达到最大。因此，可得知该问题是要在对象 o_k 给定的观测值集中选出合适的真值集合以满足式(6)，且真值集合应尽可能大。

$$\max\ \Phi'_k = \sum_{h=1}^{m} w_h \times f(V_{h,k}, V'_{*,k}) \tag{6}$$

式(6)为单个对象的多真值发现目标函数，表示最大化各数据源提供的观测值集合与该对象真值集合之间相似度的加权和，其中 w_h 为数据源 s_h 的数据质量权重。由式(5)可知，数据源质量权重由其内所有对象的相似度之和归一化所得，因此在求解真值集合过程中，可通过上次求解获得的数据源质量权重计算对象真值，然后通过本次求解获得的真值集合计算数据源质量权重。

综上可知，该多真值计算过程为典型的子集问题，即要在给定的对象值集中选出合适的真值集合以满足目标函数。因此可根据对象的数量将对象的多真值计算过程转换为同等数量的子集问题，即多子集问题。

4　MAC-SO-MTD 算法设计

由 3.2 节可知多真值计算过程是从对象观测值集中寻找到合适的真值集合，根据对象个数可转换成多子集问题。

求解子集问题是无序组合优化问题，但因子集问题的解为一个与元素次序无关的集合，与蚂蚁寻找最短觅食路径的自然行为不一致，给蚁群算法带来了挑战。文献［14］提出了基于图的蚂蚁系统，通过定义等效路径将问题本身的无序信息转换为等效路径上信息素量，并且采用基于等效路径增强的信息素更新策略对蚂蚁实施了有序影响，增加了问题求解的信息量，有效解决了蚂蚁构造解的有序性与解无序性之间的矛盾。

因此，本文在基于图的蚂蚁系统上设计蚁群算法进行求解多真值发现问题。

4.1　多蚁群同步优化的多真值发现算法的流程描述

多真值发现问题中每个对象的多真值计算过程都可转换成子集问题，据此可以设置相应数量的蚁群同步进行多真值计算，因此 MAC-SO-MTD 算法流程如图 1 所示。

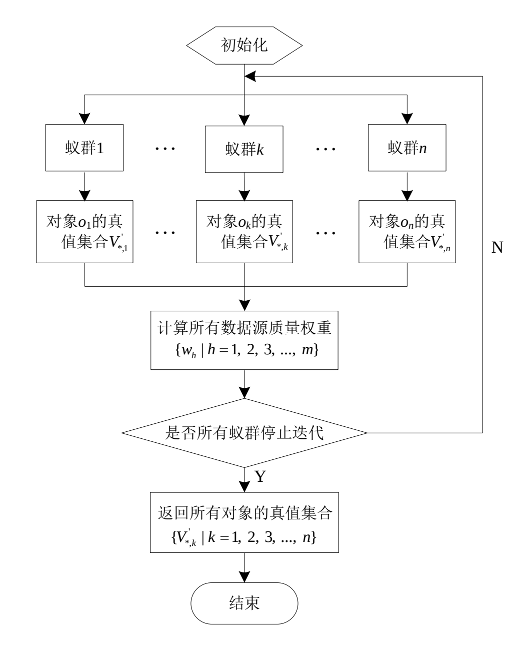


图 1　MAC-SO-MTD 算法流程

Fig.1　Flow chart of MAC-SO-MTD algorithm

图 1 中，蚁群数量根据对象个数进行设置，每个蚁群都对应一个对象，所有蚁群同步进行搜索，每次迭代完成都输出对应对象真值集合。数据源质量权重根据式(5)进行计算。MAC-SO-MTD 算法整个的计算过程是蚁群进行真值寻找和数据源质量权重计算的一个迭代过程。蚁群满足收敛条件即停止迭代，其对应对象的真值集合为历史最优解对应的真值结合。当所有蚁群停止迭代后，MAC-SO-MTD 算法输出所有对象真值集合。

本文将蚁群收敛条件设置为其对应对象历史最优目标函数值未更新的次数。当未更新的次数等于 H 时，蚁群不再进行搜索。MAC-SO-MTD 算法伪代码算法 1 所示。

算法 1　MAC-SO-MTD 算法

输入：数据源集合 S，对象集合 O。

输出：所有对象真值集合 $\{V'_{*,k} \mid k=1, 2, 3, ..., n\}$。

1.　$V'_{*,k} = V_{*,k}$，根据式(5)计算 $\{w_h \mid h=1, 2, 3, ..., m\}$；

2.　根据 $V'_{*,k}$ 及 w_h，通过式(1)计算目标函数值 Φ'；

```
3.  G = Φ', Object_k = V'_{*,k}, Source_h = w_h, Threshold 和 temp
    均为长度为 n 的零向量；
4.  生成 n 群蚂蚁并放置于对应的 U_1^k，初始化蚁群算法参
    数，蚁群收敛条件 H；
5.  while(向量 Threshold 有不为 H 的元素)
6.      for k=1 to n
7.          调用蚁群 k 寻找对象 o_k 的真值集合
            V'_{*,k}，最优函数值 G_k；
8.          if  G_k <temp[k] then
9.              Threshold[k]=Threshold[k]+1；
10.         else Threshold[k]=0, temp[k]= G_k；
11.         end if
12.     end for
13.     for h=1 to m
14.         根据式(5)计算数据源 s_h 的质量权重 w_h；
15.     end for
16.     根据式(1)计算目标函数值 Φ'；
17.     if  G < Φ'  then
18.         G = Φ', Object_k = V'_{*,k}, Source_h = w_h；
19.         根据式(10)更新 n 群蚂蚁的信息素；
20.     end if
21. end while
22. return {V'_{*,k} | k = 1, 2, 3, ..., n}
```

算法 1 中第 1~2 行是在假设所有观测值集合均为真值的基础上进行数据源质量权重和目标函数值的计算；第 5 行判断向量 ***Threshold*** 里是否存在不为 H 的元素，若存在则算法继续运行，否则返回所有对象的真值集合；第 6~12 行为蚁群进行多真值寻找步骤，返回每次迭代蚁群找到的最优值 G_k 及其对应的真值集合 $V'_{*,k}$，同时记录历史最优值未更新次数；第 13~15 行进行数据源质量权重的计算；第 16 行根据当前得到的真值集合及数据源质量权重计算目标函数值 Φ'；第 17~20 行将得到的目标函数值与目前保留的最大目标函数值进行比较，从而决定是否更新算法各参数和蚁群的信息素。当所有的蚁群满足收敛条件时，即向量 ***Threshold*** 里的元素均等于 H 时，算法退出循环，并输出所有对象真值集合。

4.2 蚁群算法设计

4.2.1 蚁群算法组成

下面为对象 o_k 对应的第 k 个蚁群算法的组成。首先构造子集问题的有向图，如图 2 所示。

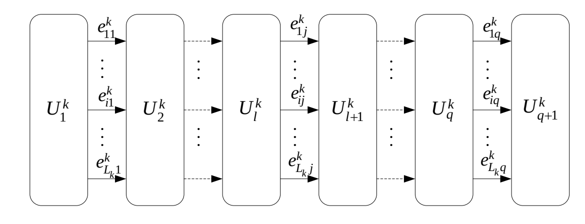


图 2 子集问题构造图的有向图

Fig.2 Directed graph of subset problem’s structure graph

图 2 中 L_k 为子集问题解的个数，即对象的真值个数；q 为蚂蚁所找解的最大可能基数；e_{ij}^k 表示第 k 个蚁群第 j 步选择第 i 个元素。

在基于图的蚂蚁系统中使用的路径选择概率公式如式(7)所示。

$$h_{ij}^k(t)=\begin{cases}\dfrac{\left(\tau_{ij}^k(t-1)\right)^{\alpha}\left(\eta_i^k\right)^{\beta}}{\sum\limits_{e_{ij}^k\notin tabu_g^k}\left(\tau_{ij}^k(t-1)\right)^{\alpha}\left(\eta_i^k\right)^{\beta}} & e_{ij}^k\notin tabu_g^k\\ 0 & \text{其他}\end{cases}\tag{7}$$

其中：禁忌表 $tabu_g^k$ (g=1, 2, …, N)记录第 k 个蚁群中第 g 只蚂蚁走过的边；α 与 β 表示信息素量和启发式因子的重要程度；$\tau_{ij}^k(t)$ 表示在 t(t=0, 1, 2, …)时刻边 e_{ij}^k 上的信息素量；启发式因子 η_i^k 是外部信息，表示选择第 k 个蚁群中第 i 个元素的希望程度，其表达式如式(8)所示。

$$\eta_i^k=\frac{\sum\limits_{h=1}^{m}\mathbf{sum}_i^k[h]}{\sum\limits_{h=1}^{m}|V_{h,k}|}\tag{8}$$

其中：$\sum\limits_{h=1}^{m}|V_{h,k}|$ 表示第 k 个对象所有观测值出现的次数之和；$\sum\limits_{h=1}^{m}\mathbf{sum}_i^k[h]$ 表示第 k 个对象的观测值集 $V_{*,k}$ 中第 i 个观测值出现的次数；向量 $\mathbf{sum}_i^k[h]$ 表达式如式(9)所示。

$$\mathbf{sum}_i^k[h]=\begin{cases}1, & v_i\in V_{h,k}\\ 0, & v_i\notin V_{h,k}\end{cases}\tag{9}$$

其中：$\mathbf{sum}_i^k[h]$ 表示数据源 s_h 为对象 o_k 提供第 i 观测值的情况，若数据源 s_h 为对象 o_k 提供了第 i 观测值，则设置为 1，否则设置为 0。

当所有蚁群一次迭代完成后，根据计算得到的目标函数值决定是否对等效路径上的信息素进行更新，信息素更新公式如式(10)所示。

$$\tau_{ij}^k(t)=\begin{cases}p(t)+\dfrac{\Phi_k'(tabu^k(t))}{Q_k} & e_{ij}^k\in\Gamma_k(tabu^k(t))\\ p(t) & \text{其他}\end{cases}\tag{10}$$

其中：$\dfrac{\Phi_k'(tabu^k(t))}{Q_k}$ 为信息素增量公式，$\Phi_k'(tabu^k(t))$ 为第 k 个蚁群中要进行信息素更新的路径的目标函数值；$\Gamma_k(tabu^k(t))$ 表示第 k 个蚁群中要进行信息素更新的等效路径[14]；Q_k 为常数，用来调整信息素增加的量；$p(t)$ 表示挥发后的信息素矩阵，如式(11)所示。

$$p(t)=(1-\rho)\times\tau_{ij}^k(t-1)\tag{11}$$

其中：ρ 为信息素挥发的系数，$0<\rho<1$。

为兼顾算法收敛速度和全局搜索能力，本文采用本次迭代最优更新和本次迭代不更新的信息素更新策略，即若本次迭代最优解好于当前全局最优解，则对本次迭代的最优路径 $tabu^k$ 进行信息素矩阵更新；若本次迭代最优解等于或小于当前全局最优解，则本次迭代不更新，以强化同等信息素分布下的搜索

力度。

4.2.2 蚁群算法的流程描述

为尽可能多地从对象观测值集中找出真值，可假设对象观测值集的元素均为真值，蚁群 k 对其进行非真值的搜索。蚁群 k 一次迭代搜索多真值的具体流程如图 3 所示。

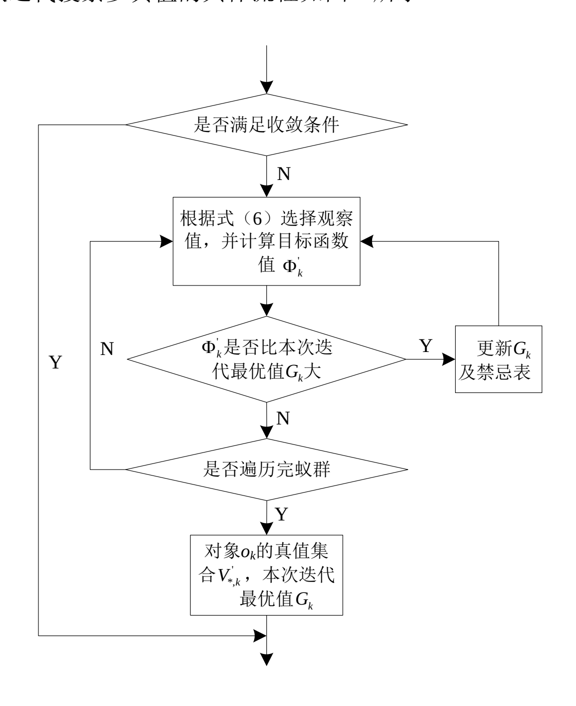


图 3　蚁群 k 一次迭代搜索多真值流程

Fig.3　Flow chart of one iterative search of multi-truth by ant colony k

图 3 中，每只蚂蚁每搜索一次就根据式(6)进行目标函数值计算，并与当前最优值进行比较，如小于当前最优值时则退出搜索。当所有蚂蚁都搜索完后，返回当前最优目标函数值对应的真值集合。当蚁群 k 满足收敛条件后不再进行多真值寻找。蚁群算法进行多真值发现的伪代码如算法 2 所示。

算法 2　蚁群搜索多真值算法

输入：数据源集合 S，对象集合 O，***Threshold***[k]，蚁群收敛条件 H，*Object*$_k$。

输出：对象真值集合 $V_{*,k}^{'}$，本次迭代最优值 G_k。

1.　*Object*$_k$，根据式(6)计算目标函数值 $\Phi_k^{'}$，　$G_k=\Phi_k^{'}$；

2.　if ***Threshold***[k]不等于 H then

3.　　for g=1 to N

4.　　　for j=1 to q

5.　　　　第 g 只蚂蚁根据式(7)在 $V_{*,k}$ 中进行冲突值的选择；

6.　　　　根据式(6)计算目标函数值 $\Phi_k^{'}$；

7.　　　　if　$\Phi_k^{'}>G_k$　then

8.　　　　　$G_k=\Phi_k^{'}$，并将该观测值加入禁忌表 $tabu_g^k$；

9.　　　　else return

10.　　　　end if

11.　　　end for

12.　　end for

13.　　根据 G_k 对应的禁忌表得到 $V_{*,k}^{'}$；

14.　　return　$V_{*,k}^{'}$，G_k

15.　end if

算法 2 中第 1 行根据对象 o_k 的历史最优值对应的真值集合 *Object*$_k$ 计算其目标函数值；第 2 行判断蚁群是否收敛，若不收敛则算法继续运行；第 5~6 行表示第 g 只蚂蚁在第 j 步寻找一个冲突值，并计算目标函数值；第 7~10 行将得到目标函数值与目前保留的最大目标函数值进行比较，以此判断第 g 只蚂蚁是否继续寻找真值；第 13 行根据本次迭代最优值 G_k 对应的禁忌表计算真值集合 $V_{*,k}^{'}$；第 14 行返回每次迭代的最优值 G_k 及其对应的真值集合 $V_{*,k}^{'}$。

4.3　算法复杂度分析

4.3.1 时间复杂度

算法 2 各步骤的时间复杂度（最坏情况）如表 2 所示。

表 2　算法 2 各步骤时间复杂度

Table 2　Time complexity of each step of algorithm 2

步骤	时间复杂度
蚂蚁禁忌表	$O(N\times L_k)$
蚂蚁可搜索可行路径	$O(N\times L_k^2)$
解的评价更新	$O(N)$

算法 2 中蚁群只负责找出对象的真值集合，N 为蚁群中蚂蚁数量，对问题规模没有影响，因此由表 2 可知算法 2 的时间复杂度为 $O(L_k^2)$。

算法 1 各步骤的时间复杂度（最坏情况）如表 3 所示。

表 3　算法 1 各步骤时间复杂度

Table 3　Time complexity of each step of algorithm 1

步骤	时间复杂度
初始化蚁群参数（$\tau_{ij}^k(0)$，η_i^k）	$O(L_k^2+L_k)$
算法 2	$O(H\times n\times L_k^2)$
数据源权重的计算	$O(H\times m)$
信息素更新	$O(H\times n\times L_k^2)$
解的评价更新	$O(H\times N)$

由表 3 可知，MAC-SO-MTD 算法时间复杂度为 $O(H\times n\times L_k^2)$。

4.3.2 空间复杂度

算法 2 实际实现时，有向图的功能可以由信息素表兼任，并不占用存储空间。算法 2 各部分的空间复杂度如表 4 所示。

表 4　算法 2 各部分空间复杂度

Table 4　Space complexity of each part of algorithm 2

存储内容	空间复杂度
物品价值	$O(L_k)$
信息素表 $\tau_{ij}^k(0)$	$O(L_k^2)$
启发式因子 η_i^k	$O(L_k)$
禁忌表	$O(L_k)$
数据源质量权重	$O(m)$
数据源集合	$O(m)$
对象集合	$O(n)$
对象观测值集合	$O(m\times L_k)$
对象真值集合	$O(L_k)$

算法 2 中蚁群只负责找出对象的真值集合，因此由表 4 可知算法 2 的空间复杂度为 $O(L_k^2)$。

算法 1 各部分的空间复杂度如表 5 所示。

表 5　算法 1 各部分空间复杂度

Table 5　Space complexity of each part of algorithm 1

存储内容	空间复杂度
算法 2	$O(n\times L_k^2)$
数据源质量权重	$O(m)$
信息素表 $\tau_{ij}^k(0)$	$O(n\times L_k)$
向量 ***Threshold***	$O(n)$
向量 ***temp***	$O(n)$
数据源集合	$O(m)$
对象集合	$O(n)$
对象观测值集合	$O(n\times m\times L_k)$
对象真值集合	$O(n\times L_k)$

由表 5 可知，MAC-SO-MTD 算法空间复杂度为 $O(n\times L_k^2)$。

5　实验与分析

本章通过在真实数据集上进行对比实验，验证了多蚁群同步优化的多真值发现算法的有效性和准确性。

5.1　实验数据及方法

本文所提算法解决的是多真值发现问题，因此实验采用的数据集应具有多值属性，如图书的作者属性、电影的导演属性等。本文采用两个真实数据集：a)Books-Authors 数据集，包含多个网站提供的图书和作者的信息；b)Movies-Directors 数据集，包含多个电影网站提供的电影和导演的信息。

a) Books-Authors 数据集[3]。该数据集是真值发现算法常用的数据集，其中包括 877 个数据源、1 263 本书籍以及 33 971 条记录，且其提供了 100 本书籍作者的真实信息。本文去掉原始数据集中的重复记录和无作者信息的记录，并对作者姓与名进行分割，经过处理后的数据集包含 877 个数据源、1 263 本书籍以及 25 604 条记录，其中作者可能值集大小为[1, 54]，平均可能值集大小为 7.7。该数据集的标准集为随机挑选出 100 本图书并对其作者信息进行手工标注后的记录。

b) Movies-Directors 数据集[12]。一部电影的导演可以有多个，因此电影的导演属性是一个多真值属性。Movies-Directors 数据集包含了 15 个国外电影网站的各种类电影 468 607 部，共 1 134 432 条记录。本文根据电影上映年份，抽取 2010—2017 年间上映电影的记录，经去掉重复记录和无导演信息的记录，并对导演姓与名进行分割，最终得到的数据集包含 15 个数据源、36 242 部电影以及 104 591 条记录。其中导演可能值集大小为[1, 71]，平均可能值集大小为 3.1。该数据集的标准集为随机挑选出 188 部电影并对其导演信息进行手工标注后的记录。

将本文所提方法分别与 Voting 算法和 Mtruths_Greedy 算法和遗传算法（genetic algorithm, GA）[15]进行对比，设置如下：

方法 1　Voting-K。该算法采用投票机制计算真值，本文选择投票比重大于 K 的作为真值。

方法 2　Mtruths_Greedy。该算法是 Mtruths 算法提出的一种算法，在真值计算过程中采用贪心策略来判断真值集合。

方法 3　MGA-MTD。该方法框架与本文所提算法框架相同，其中多真值寻找过程采用经典遗传算法同步进行搜索，算法停止条件与 MAC-SO-MTD 算法一致。其中 Books-Authors 数据集中 MGA-MTD 算法参数设置：交叉率为 0.5，变异率为 0.01，染色体个数为 30；Movies-Directors 数据集中 MGA-MTD 算法参数设置：交叉率为 0.6，变异率为 0.01，染色体个数为 50。

方法 4　本文第 4 章提出的 MAC-SO-MTD 算法。该算法采用多蚁群同步进行寻找真值集合，其中根据文献［14,16］和结合数据集特点，Books-Authors 数据集中 MAC-SO-MTD 算法参数设置为：信息素初始化浓度 $\tau_{ij}(0)=100$，信息素重要程度 $\alpha=0.8$，启发式信息重要程度 $\beta=0.65$，信息素挥发系数 $\rho=0.1$，常数 Q=20，蚂蚁个数 N=15；Movies-Directors 数据集中 MAC-SO-MTD 算法参数设置为：信息素初始化浓度 $\tau_{ij}(0)=100$，信息素重要程度 $\alpha=1$，启发式信息重要程度 $\beta=0.6$，信息素挥发系数 $\rho=0.1$，常数 Q=400，蚂蚁个数 N=20。

本文实验采用 MATLAB 实现所有算法，软件开发环境为 MATLAB R2017a。实验的内存大小为 16 GB，处理器为 Intel(R) Core(TM) i7-4770，采用 Windows7 64 位操作系统。

5.2　评价指标

实验结果采用文献［13］中的衡量真值发现算法准确性的三个指标来衡量算法的优劣。

1）查准率（precision）　表示算法得到对象的真值集合中正确真值所占的比例，计算公式如式(12)所示。

$$P=\frac{|T_k\cap V_{*,k}^{'}|}{|V_{*,k}^{'}|}\times 100\% \tag{12}$$

其中：T_k 表示对象 o_k 所有的真值集合；$V_{*,k}^{'}$ 表示算法得到的对象 o_k 的真值集合；$T_k\cap V_{*,k}^{'}$ 表示算法得到的真正为对象 o_k 真值的集合。

2）查全率（recall）　表示算法得到的真值集合中正确真值占对应正确真值集的比例，计算公式如式(13)所示。

$$R=\frac{|T_k\cap V_{*,k}^{'}|}{|T_k|}\times 100\% \tag{13}$$

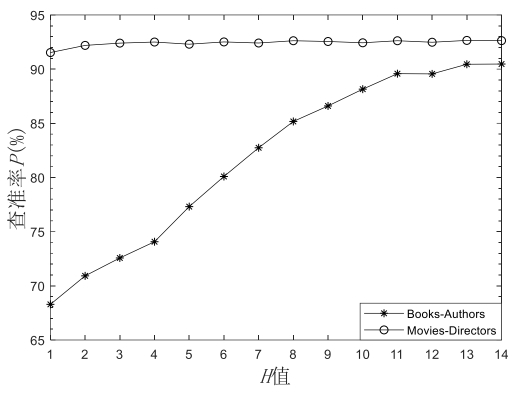
3）$F1$ 指标　表示查准率和查全率的调和平均数，计算公式如式(14)所示。

$$F1=\frac{2\times P\times R}{P+R} \tag{14}$$

5.3　MAC-SO-MTD 算法参数敏感性分析

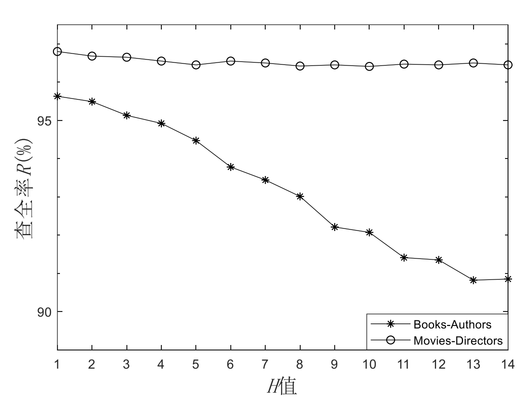
MAC-SO-MTD 算法中采用目标函数值未更新次数 H 作为蚁群收敛的条件，因此需要对 H 值的敏感性进行分析。

在 Books-Authors 数据集与 Movies-Directors 数据集上对 H 值取不同值，分别运行 10 次计算其对应 P、R 和 $F1$ 的均值，实验结果如图 4 所示。



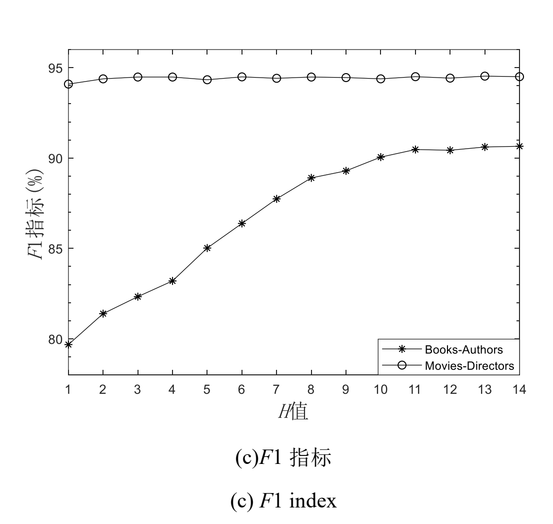
(a) 查准率 *P*

(a) Precision rate *P*



(b) 查全率 *R*

(b) Recall ratio *R*



(c)*F*1 指标

(c) *F*1 index

图 4　*H* 值对查准率 *P*、查全率 *R* 和 *F*1 指标的影响

Fig.4　Influence of *H* value on precision *P*, recall *R* and *F*1

由图 4 可看出，在 Books-Authors 数据集上，MAC-SO-MTD 算法的查准率 *P* 和 *F*1 指标随 *H* 值的增大而增大，查全率 *R* 随 *H* 值的增大而逐渐减小。当 *H* 值大于等于 11 时，MAC-SO-MTD 算法的 *F*1 指标趋于稳定，因此在 Books-Authors 数据集中蚁群的收敛条件可设置为对应目标函数值未更新 11 次。

而在 Movies-Directors 数据集上，MAC-SO-MTD 算法的查准率 *P*、查全率 *R* 和 *F*1 指标波动性较小。当 *H* 值大于等于 6 时 *F*1 指标趋于稳定，因此在 Movies-Directors 数据集中蚁群的收敛条件可设置为对应目标函数值未更新 6 次。

5.4　对比结果分析

本文所提算法分别对比于 Voting 算法和、Mtruths_Greedy 算法和 MGA-MTD 算法。其中 Voting 算法为真值发现的基准算法；Mtruths_Greedy 算法可以直接返回对象的真值集合，且在准确性方面优于现有多真值算法；MGA-MTD 算法采用遗传算法进行多真值的寻找。

原始的 Voting 算法根据投票的多少给出观测值为真的可能性，不能直接返回真值集合，因此需要设置一个阈值 *K*，选择概率大于 *K* 的观测值为真值。实验设置 *K* 值分别为 15%、30%、45%。

在两个数据集上分别用 Voting-*K*、Mtruths_Greedy、MGA-MTD 及 MAC-SO-MTD 算法进行实验，运行 10 次计算 *P*、*R* 和 *F*1 的均值和标准差，实验结果如表 6、7 所示。

表 6　不同方法在 Books-Authors 数据集上的真值发现实验结果

Table 6　Results of different truth discovery methods on the Book-Authors data set

方法	*P*	*R*	*F*1
Voting-15%	98.57±0	62.64±0	76.60±0
Voting-30%	100±0	19.82±0	33.08±0
Voting-45%	100±0	5.47±0	10.37±0
Mtruths_Greedy	89.85±0	80.64±0	85.00±0
MGA-MTD	89.7±0.93	90.82±0.83	90.25±0.66
MAC-SO-MTD	89.58±1.66	91.41±0.80	90.47±0.68

表 7　不同方法在 Movies-Directors 数据集上的真值发现实验结果

Table 7　Results of different truth discovery methods on Movies-Directors data set

方法	*P*	*R*	*F*1
Voting-15%	98.29±0	88.31±0	93.04±0
Voting-30%	99.07±0	60.92±0	75.45±0
Voting-45%	98.66±0	42.34±0	59.25±0
Mtruths_Greedy	95.78±0	91.19±0	93.43±0
MGA-MTD	90.54±0.3	97.17±0.16	93.74±0.15
MAC-SO-MTD	92.52±0.12	96.47±0.08	94.45±0.08

由表 6 与 7 可知，MAC-SO-MTD 算法的查全率 *R* 与 *F*1 指标均优于 Mtruths_Greedy 算法，且 *F*1 指标优于 Voting-*K* 算法与 MGA-MTD 算法；而 Voting-*K* 算法的查准率 *P* 虽然稍高于其他对比算法，但由于其查全率 *R* 较低，所以 Voting-*K* 算法的 *F*1 指标明显低于其他对比算法。在 Books-Authors 数据集中，MGA-MTD 算法的查准率 *P* 虽然稍高于 MAC-SO-MTD 算法，但其查全率 *R* 与 *F*1 指标均低于 MAC-SO-MTD 算法。而在 Movies-Directors 数据集中，MAC-SO-MTD 算法的 *F*1 指标显著高于其他三种算法。

通过在两个真实数据集上的实验可知，MAC-SO-MTD 算法的准确性比 MGA-MTD 算法更高。而 Voting-*K* 算法中由于占比越高的观测值越可能为真值，随着阈值 *K* 的增大，查准率

P 越大，但其查全率 R 显著降低，因此无法返回对象完整的真值集合。Mtruths_Greedy 算法是基于贪心策略进行多真值发现，将对象观测值集里的观测值按可能为真值的概率进行排列并挑选，当对象的真值较多且分布较均匀时易陷入局部最优，故其查全率 R 和 $F1$ 指标均低于 MAC-SO-MTD 算法。

6 结束语

数据在各行各业中发挥着越来越重要的作用，如何从冲突数据中挖掘出准确的数据具有重要的意义和研究价值。真值发现作为数据集成中冲突消解的有效手段，得到了广泛研究。然而，当前的研究工作更多地关注单真值发现问题。针对多真值发现问题，本文提出了一种基于多蚁群同步优化的多真值发现算法 MAC-SO-MTD，将对象的多真值发现过程转换成子集问题，并设计多蚁群算法同步进行真值搜索，避免了阈值选择的问题，提高了多真值发现的准确性，在对象真值集合基数较大时能较好地进行多真值发现。同时，考虑了数据源权重对真值发现效果的影响，在计算过程中迭代地进行蚁群真值寻找和数据源质量权重计算。

参考文献：

[1] Bleiholder J, Naumann F. Data fusion [J]. ACM Computing Surveys, 2008, 41 (1): 1-41.

[2] 曹建军，刁兴春，汪挺，等. 领域无关数据清洗研究综述 [J]. 计算机科学, 2010, 37 (5): 26-29. (Cao Jianjun, Diao Xingchun, Wang Ting, *et al.* Research on domain-independent data cleaning: a survey [J]. Computer Science, 2010, 37 (5): 26-29.)

[3] Yin Xiaoxin, Han Jiawei, Yu Philip S. Truth discovery with multiple conflicting information providers on the web [J]. IEEE Trans on Knowledge and Data Engineering, 2008, 20 (6): 796-808.

[4] 考明军，张炜，高宏. 冲突数据中的真值发现算法 [J]. 计算机研究与发展, 2010, 47 (z1): 188-192. (Kao Mingjun, Zhang Wei, Gao Hong. Truth discovery methods in conflict data integration [J]. Journal of Computer Research and Development, 2010, 47 (z1): 188-192.)

[5] Li Qi, Li Yaliang, Gao Jing, *et al.* Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation [C]// Proc of the 32nd ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2014: 1187-1198.

[6] Li Qi, Li Yaliang, Gao Jing, *et al.* A confidence-aware approach for truth discovery on long-tail data [J]. Proceedings of the VLDB Endowment, 2014, 8 (4): 425-436.

[7] Li Yaliang, Li Qi, Gao Jing, *et al.* Conflicts to harmony: a framework for resolving conflicts in heterogeneous data by truth discovery [J]. IEEE Trans on Knowledge and Data Engineering, 2016, 28 (8): 1986-1999.

[8] Dong Xinluna, Gabrilovich E, Murphy K, *et al.* Knowledge-based trust: estimating the trustworthiness of web sources [J]. Proceedings of the VLDB Endowment, 2015, 8 (9): 938-949.

[9] 余东，申德荣，寇月，等. 面向 Web 数据集成的真值发现算法 [J]. 小型微型计算机系统, 2016, 37 (8): 1633-1638. (Yu Dong, Shen Derong, Kou Yue, *et al.* Web data integration oriented truth discovery algorithms [J]. Journal of Chinese Computer Systems, 2016, 37 (8): 1633-1638.)

[10] Zhao Bo, Rubinstein B I P, Gemmell J, *et al.* A Bayesian approach to discovering truth from conflicting sources for data integration [J]. Proceedings of the VLDB Endowment, 2012, 5 (6): 550-561.

[11] 王继奎，李少波. 基于 HITS 的冲突 Deep Web 数据多真值发现算法 [J]. 计算机工程, 2016, 42 (9): 158-162. (Wang Jikui, Li Shaobo. Multiple truth value discovery algorithm for conflicting Deep Web data based on HITS [J]. Computer Engineering, 2016, 42 (9): 158-162.)

[12] Lin Xueling, Chen Lei. Domain-aware mutil-truth discovery from conflicting sources [J]. Proceedings of the VLDB Endowment, 2018, 11 (5): 635-647.

[13] 马如霞，孟小峰，王璐，等. MTruths: Web 信息多真值发现方法 [J]. 计算机研究与发展, 2016, 52 (12): 2858-2866. (Ma Ruxia, Meng Xiaofeng, Wang Lu, *et al.* MTruths: an approach of multiple truths finding from web information [J]. Journal of Computer Research and Development, 2016, 52 (12): 2858-2866.)

[14] 曹建军，张培林，王艳霞，等. 一种求解子集问题的基于图的蚂蚁系统 [J]. 系统仿真学报, 2008, 20 (22): 6146-6150. (Cao Jianjun, Zhang Peilin, Wang Yanxia, *et al.* Graph-based ant system for subset problems [J]. Journal of System Simulation, 2008, 20 (22): 6146-6150.)

[15] 王小平，曹立明. 遗传算法理论应用与软件实现 [M]. 西安：西安交通大学出版社, 2002: 18-50. (Wang Xiaoping, Cao Liming. Genetic algorithm: theory, application and software implementation [M]. Xi’an: Xi’an Jiaotong University Press, 2002: 18-50.)

[16] Dorigo M, St ü tzle T. 蚁群优化 [M]. 张军，胡晓敏，罗旭耀，等译. 北京：清华大学出版社, 2007: 202-209. (Dorigo M, St ü tzle T. Ant colony optimization [M]. Zhang Jun, Hu Xiaomin, Luo Xuyao, et al Translated. Beijing: Tsinghua University Press, 2007: 202-209.)